

**Claims:**

1           **1.**     A method for maintaining a connection between a server and a  
2 client comprising the steps of:  
3                 receiving a message;  
4                 determining whether to store one or more elements of the  
5 message;  
6                 storing the one or more elements of the message when the one or  
7 more elements of the message are to be stored;  
8                 transmitting the message; and  
9                 determining whether the server has failed and when the server has  
10 failed, restoring the server to a pre-failure connection state using the one or more  
11 stored message elements.

1           **2.**     The method as recited in claim **1**, further comprising the steps of:  
2                 determining whether to delay the message; and  
3                 delaying the transmission of the message until one or more  
4 conditions are satisfied when the message is to be delayed

1           **3.**     The method as recited in claim **2**, wherein the one or more  
2 conditions comprises successful storage of the one or more elements of the  
3 message.

1           **4.**     The method as recited in claim **1**, further comprising the steps of:  
2                 determining whether to discard the message;  
3                 discarding the message when the message is to be discarded; and  
4                 wherein the message is not transmitted when the message is to be  
5 discarded.

1           **5.**     The method as recited in claim **1**, further comprising the steps of:  
2                 determining whether to modify the message; and

3                    modifying one or more elements of the message when the  
4                    message is to be modified.

1            **6.**        The method as recited in claim **1**, wherein the step of restoring the  
2                    application to a pre-failure connection state using the one or more stored  
3                    message elements further comprises periodically transmitting an outgoing  
4                    message to maintain the connection until the application is restored.

1            **7.**        The method as recited in claim **1**, further comprising the step of  
2                    periodically storing a current state of the server and discarding any stored  
3                    elements that are no longer needed to restore the server to the current state.

1            **8.**        The method as recited in claim **1**, wherein the message is a  
2                    protocol segment.

1            **9.**        The method as recited in claim **1**, wherein the message conforms  
2                    to the Transmission Control Protocol (TCP) standard.

1            **10.**       The method as recited in claim **1**, wherein the one or more  
2                    elements are stored in a log server

1            **11.**       The method as recited in claim **10**, wherein the log server is  
2                    remotely located from the server.

1            **12.**       The system as recited in claim **1**, wherein the one or more  
2                    elements are stored in a stable memory in the server.

1            **13.**       The method as recited in claim **1**, wherein the one or more  
2                    elements are stored in a secondary server.

1            **14.**       The method of claim **1**, further including the step of periodically  
2                    transmitting, when said server has failed, an outgoing message to the client to  
3                    maintain the connection until the server is restored.

1           **15.**   A system comprising:  
2                   a server computer having a process layer, a protocol layer coupled  
3 to the process layer through a first wrapper, and a network interface coupled to  
4 the protocol layer through a second wrapper;  
5                   a log server coupled to the first wrapper and the second wrapper;  
6                   one or more client computers; and  
7                   one or more network connections between the one or more client  
8 computers and the network interface of the server computer.

1           **16.**   The system as recited in claim **15**, wherein the first wrapper is  
2 interposed in a first interface between the process layer and the protocol layer,  
3 and the second wrapper is interposed in a second interface between the protocol  
4 layer and the network interface.

1           **17.**   The system as recited in claim **15**, wherein the first wrapper and the  
2 second wrapper are interposed in the protocol layer.

1           **18.**   The system as recited in claim **15**, wherein the first wrapper and  
2 second wrapper are capable of receiving a message, determining whether to  
3 store one or more elements of the message, storing the one or more elements of  
4 the message in the log server when the one or more elements of the message  
5 are to be stored, transmitting the message, and determining whether the server  
6 has failed and when the server has failed, restoring the server to a pre-failure  
7 connection state using the one or more stored message elements.

1           **19.**   The system as recited in claim **18**, wherein the first wrapper and  
2 second wrapper are further capable of determining whether to delay the  
3 message, and delaying the transmission of the message until one or more  
4 conditions are satisfied when the message is to be delayed.

1           **20.**    The system as recited in claim **19**, wherein the one or more  
2 conditions comprises successful storage of the one or more elements of the  
3 message.

1           **21.**    The system as recited in claim **18**, wherein the first wrapper and  
2 second wrapper are further capable of determining whether to discard the  
3 message, discarding the message when the message is to be discarded, and not  
4 transmitting the message when the message is to be discarded.

1           **22.**    The system as recited in claim **18**, wherein the first wrapper and  
2 second wrapper are further capable of determining whether to modify the  
3 message, modifying one or more elements of the message when the message is  
4 to be modified.

1           **23.**    The system as recited in claim **18**, wherein the second wrapper is  
2 further capable of periodically transmitting an outgoing message to maintain the  
3 connection until the application is restored.

1           **24.**    The system as recited in claim **18**, wherein the first wrapper and  
2 second wrapper are further capable of periodically storing a current state of the  
3 server and discarding any stored elements that are no longer needed to restore  
4 the server to the current state.

1           **25.**    The system as recited in claim **18**, wherein the message is a  
2 protocol segment.

1           **26.**    The system as recited in claim **18**, wherein the message conforms  
2 to the Transmission Control Protocol (TCP) standard.

1           **27.**    The system as recited in claim **15**, wherein the log server is  
2 remotely located from the server.

1           **28.**    The system as recited in claim **15**, wherein the log server is a  
2 stable memory in the server.

- 1           **29.**    The system as recited in claim **15**, wherein the log server is a
- 2   secondary server.